信息检索及倒排技术

宋老师
2020/08/30

贪心科技 GREEDY TECHNOLOGY
NeuHub AI Links All

提纲

- **信息检索概述**
- 布尔检索及倒排索引
- 信息检索模型综述

- 索引压缩与VB编码
- 查询优化与WAND算法
- 信息检索系统评估

NeuHub
AI Links All

信息检索是什么

- Information Retrieval这个术语产生于 Calvin Mooers 1948年在MIT的硕士论文。

- Information Retrieval(IR)：从大规模非结构化数据（通常是文本）的集合（通常保存在计算机上）中找出满足用户信息需求的资料（通常是文档）的过程。

- 作为一门学科，是研究信息的获取(acquisition)、表示(representation)、存储(storage)、组织(organization)和访问(access)的一门学问。

NeuHub AI LIKE AII

信息检索是什么

IR不仅仅是搜索，IR系统也不仅仅是搜索引擎

- 例1：返回与信息检索相关的网页->搜索引擎(SearchEngine,SE)
- 例2：曾哥是狮子座的吗？->问答系统(Question Answering,QA)
- 例3：返回Ipad的各种型号、配置、价格等->信息抽取（Information Extraction, IE)
- 例4：使用Google Reader订阅新闻，并获取推荐->信息过滤(Information Filtering)、信息推荐(Information Recommending)

信息检索的历史

- **计算机出现以前：**
  - 约4000年前，人类就开始有目的地组织信息，一个典型的例子就是图书中的目录。
  - 随后，逐渐出现索引的概念，即从一些词和概念指向相关信息或者文档的指针。
  - 计算机问世以前，人们主要通过手工方式来建立索引。

  - 例子：词典（拼音检字、部首笔画检字等）

NeuHub
AI Links All

信息检索的历史

- 1948：
  - C. N. Mooers在其MIT的硕士论文中第一次创造了“Information Retrieval”这个术语。
- 1960－70年代：
  - 人们开始使用计算机为一些小规模科技和商业文献的摘要建立文本检索系统。
  - 产生了布尔模型(Boolean Model)、向量空间模型(Vector Space Model)和概率检索模型(Probabilistic Model)。康奈尔大学的Salton领导的研究小组是该领域研究的佼佼者。
  - 伦敦城市大学的Robertson及剑桥大学的Sparck Jones是概率模型的倡导者。

信息检索的历史

- 1980年代：
  - 出现了一些商用的较大规模数据库检索系统
  - Lexis-Nexis
  - Dialog
  - MEDLINE

NeuHub
AI Links All

信息检索的历史

- 1986：Internet正式形成
- 1990’s:
  - 第一个网络搜索工具：1990年加拿大蒙特利尔麦吉尔(McGill)大学开发的FTP搜索工具Archie
  - 第一个WEB搜索引擎：1994年美国CMU开发的Lycos
  - 1995：斯坦福大学博士生开发的Yahoo
  - 1998：斯坦福大学博士生开发的Google，提出PageRank计算公式。
  - 1998：基于语言模型的IR模型提出。

京东智联云 & AI
NeuHub AI Links All

信息检索的历史

- 1990年代的其他重要事件：
  - 评测会议
    - NIST TREC
  - 推荐系统的出现
    - Ringo
    - Amazon
    - NetPerceptions
- 文本分类和聚类的使用

TREC(Text Retrieval Conference)

它是文本检索领域人气最旺、最权威的评测会议。该会议细分为几大主要方向：问题回答（QA）、特定领域检索（Legal、Genomics、Enterprise、Blog）、传统Web检索等。会议负责组织收集并向与会者提供标准的语料库（Corpus）、检索条件和问题集（Query Set）、以及评测办法（Evaluation），与会者则被要求在规定的时间内构造检索系统并提交检索结果（Runs），由会议负责评测各个检索结果的优劣，最终依据评测结果召开大会进行学术交流，发表会议论文。

信息检索的历史

- 2000’s
  - 信息抽取
    - Whizbang
    - Fetch
    - Burning Glass
  - 问答系统
    - TREC Q/A track
  - 2000年，百度成立

NeuHub
AI Links All

信息检索的历史

- 2000以来的其他重要事件：
  - 多媒体 IR
    - Image
    - Video
    - Audio and music
  - 跨语言 IR
    - DARPA Tides
  - 文本摘要
    - DUC评测

NeuHub
AI Links All

信息检索小结

- 历史悠久，经久不衰
- 演化出了更多的NLP任务，不仅仅是搜索引擎

NeuHub
AI Links All

提纲

- 信息检索概述
- **布尔检索及倒排索引**
  - 信息检索例子、倒排索引、布尔查询模型
- 信息检索模型综述

- 索引压缩与VB编码
- 查询优化与WAND算法
- 信息检索系统评估

NeuHub
AI Links All

信息检索例子

一个信息检索的例子

- 很多人都有《莎士比亚全集》这本大部头的书。我想知道：哪些剧本包含Brutus和Caesar但是不包含Calpurnia？
- 一种方式是采用Unix下的grep程序，先找出所有包含Brutus和Caesar的剧本，然后在将包含Calpurnia的剧本排除
- 但是很多情况下，采用上述线性扫描的方式是远远不够的。为什么？

信息检索例子

一个信息检索的例子

- 上述方式不合适的原因：
  1. 对于大规模文档的搜索太慢
  2. 有时我们需要更灵活的匹配方式。比如，要求查找countrymen附近的Romans。
  3. 我们需要对结果进行排序。用户希望能在多个能满足自己要求的文档中得到最佳答案。

NeuHub AI Links All

信息检索例子

词项文档索引

词项—文档关联矩阵（incidence matrix）

- 解决方法是采用非线性的扫描方式，一种方法就是事先给文档建立索引（**index**）

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	1	1	0	0	0	1
Brutus	1	1	0	1	0	0
Caesar	1	1	0	1	1	1
Calpurnia	0	1	0	0	0	0
Cleopatra	1	0	0	0	0	0
mercy	1	0	1	1	1	1
worser	1	0	1	1	1	0

如果文档（这里就是剧本）包含某个词，则对应的项为1，否则为0

信息检索例子

处理查询的过程

- 分别取出Brutus\Caesar以及Calpurnia对应的行向量，并对Calpurnia对应的向量求反。
- 进行按位与操作

  110100 *AND* 110111 *AND* 101111 = 100100.
- 结果向量中第一个和第四个元素为1，表明查询结果对应的剧本是 Antony and Cleopatra 和 Hamlet

信息检索例子

更大的数据集

- 假设有一个更大的数据集：共有N=100万个文档，每个文档包含大概1000个词汇
- 假设一个词汇需要6个字节（将空格和标点计算在内），则这些文档共有6G
- 假设其中共有500k个唯一的不重复的词汇

- 在这种情况下，前面所使用的词项文档矩阵会如何？

Links All

信息检索例子

无法构建矩阵

- 矩阵中有 500K * 1M = 0.5T 个 0和1
- 但是其中只有不到10亿个1，其他都是0.这个矩阵十分稀疏。
- 更好的索引表示方法是？
  - 只记录1的位置

	T_1	T_2	….	T_t
D_1	0	0	…	1
D_2	0	0	…	0
:	:	:		:
:	:	:		:
D_n	0	1	…	0

NeuHub AI Links All

信息检索例子

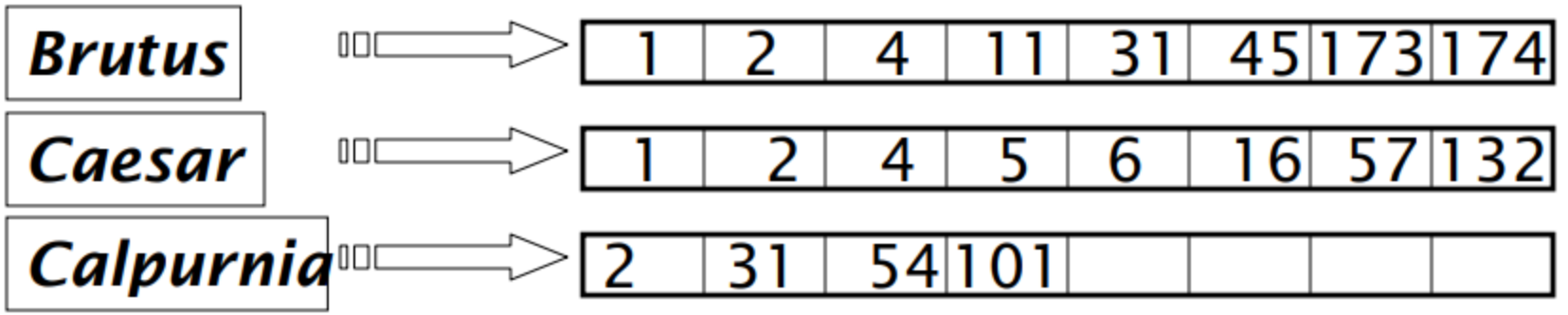
小结：一个简单的搜索示例

- 线性扫描的搜索：**grep**
- 非线性的扫描方式：**index**
- 处理查询
  - 构造矩阵 → **110100 AND 110111 AND 101111 = 100100**
- 简单模型存在的问题：大的数据集无法构建矩阵

NeuHub
AI Links All

倒排索引

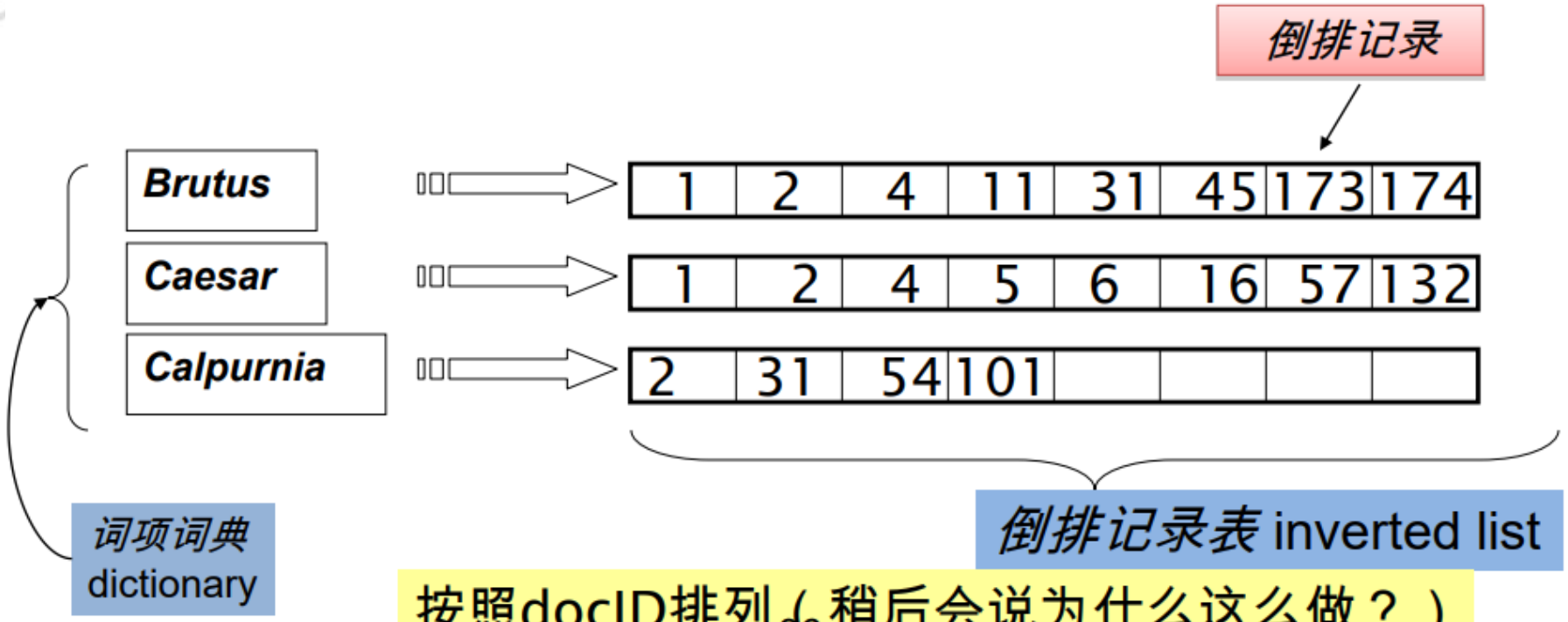
- 对于每一个词项，存储所有包含这个词项的文档的一个列表。一个文档用一个序列号**docID**来表示。
- 我们能用一个固定长度的数组来存储它吗？



如果词项*Caesar* 被添加到14号文档中，索引如何变化?

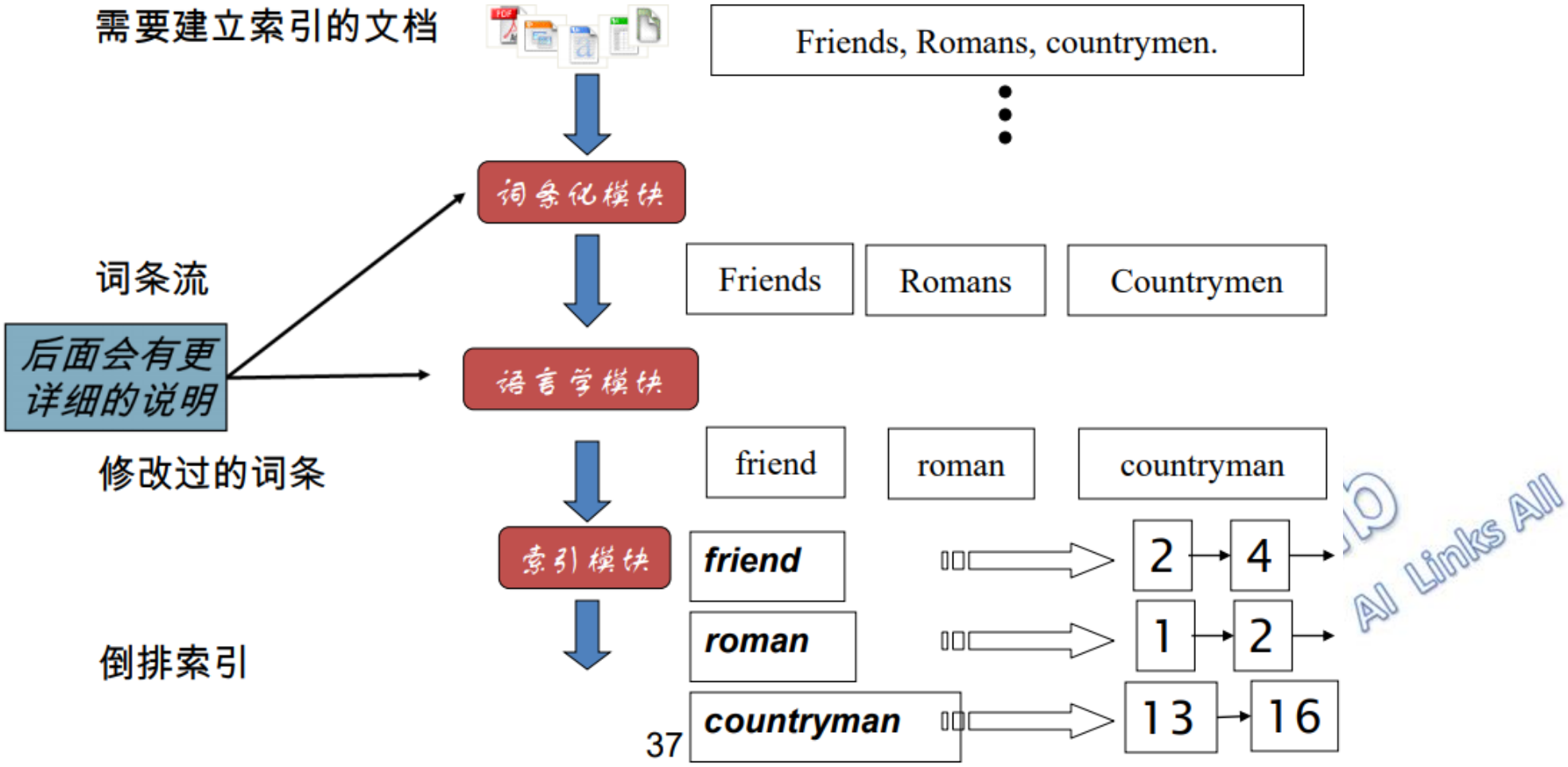
倒排索引

- 应当使用可变长度的记录列表
  - 在硬盘上，一串连续的记录是正常的，也是最好的。
  - 在内存里，可以使用链表，或者可变长度的数组。



贪心科技 | 让每个人享受个性化教育服务

倒排索引的建立


需要建立索引的文档
Friends, Romans, countrymen.
词条化模块
词条流
Friends
Romans
Countrymen
后面会有更详细的说明
语言学模块
修改过的词条
friend
roman
countryman
索引模块
倒排索引
friend
2
4
roman
1
2
countryman
13
16
37

建立索引步骤：词条序列

- （修改过的词条，文档ID）对应的序列

Doc 1

I did enact Julius Caesar I was killed i' the Capitol; Brutus killed me.

Doc 2

So let it be with Caesar. The noble Brutus hath told you Caesar was ambitious

Term	docID
I	1
did	1
enact	1
julius	1
caesar	1
I	1
was	1
killed	1
i'	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2
let	2
it	2
be	2
with	2
caesar	2
the	2
noble	2
brutus	2
hath	2
told	2
you	2
caesar	2
was	2
ambitious	2

建立索引步骤：排序

先按照词条排序，
再按照docID排序

建立索引的核心步骤

Term	docID
I	1
did	1
enact	1
julius	1
caesar	1
I	1
was	1
killed	1
i'	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2
let	2
it	2
be	2
with	2
caesar	2
the	2
noble	2
brutus	2
hath	2
told	2
you	2
caesar	2
was	2
ambitious	2

Term	docID
ambitious	2
be	2
brutus	1
brutus	2
capitol	1
caesar	1
caesar	2
caesar	2
did	1
enact	1
hath	1
I	1
I	1
i'	1
it	2
julius	1
killed	1
killed	1
let	2
me	1
noble	2
so	2
the	1
the	2
told	2
you	2
was	1
was	2
with	2

智联云 & AI Links All

建立索引步骤：词典和倒排表

- 同一篇文档中多次出现的词被合并
- 分割成词典和倒排表
- 词汇的文档频率也被记录

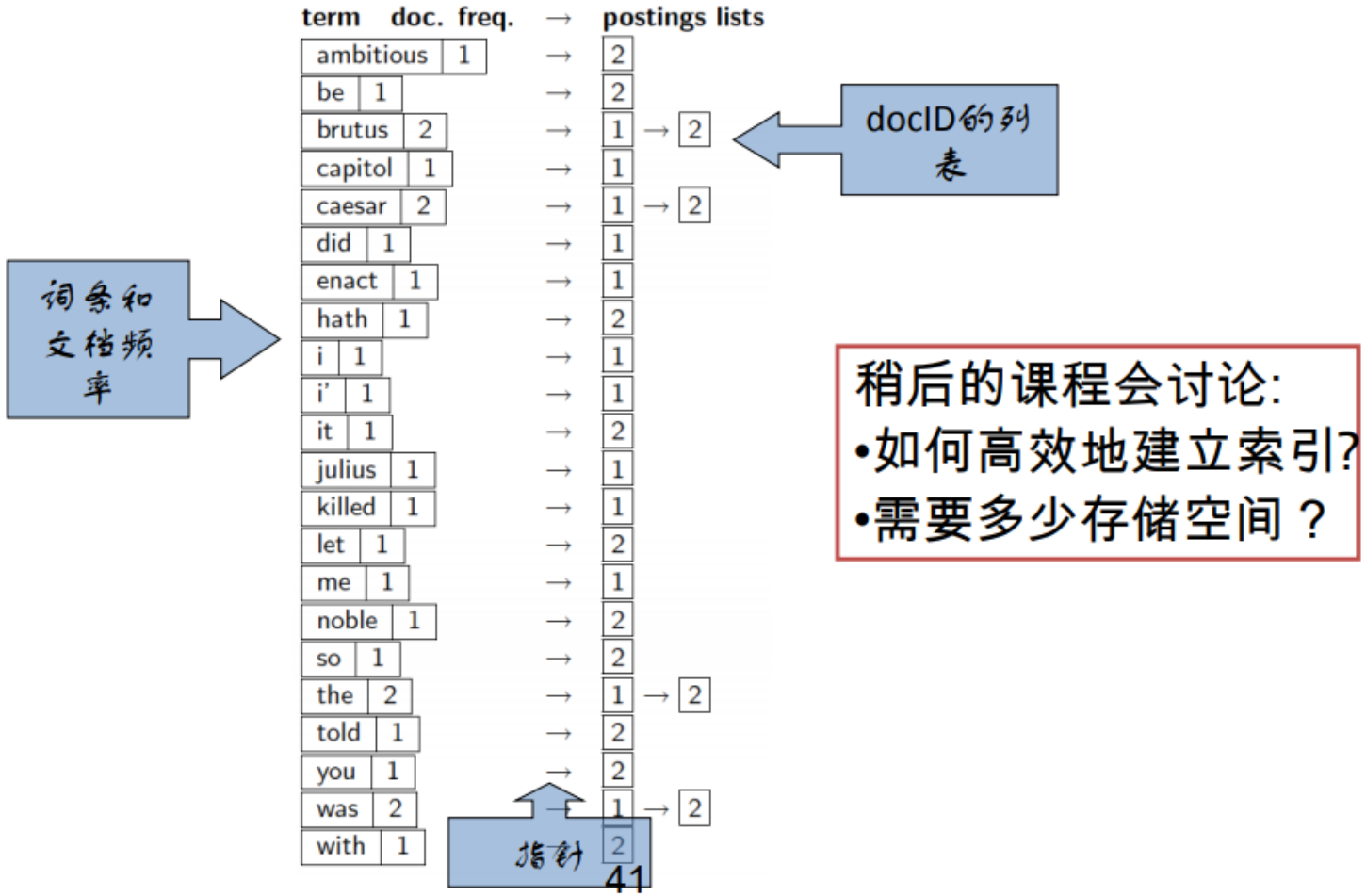
后面会讨论为什么记录文档频率

Term	docID
ambitiou	2
be	2
brutus	1
brutus	2
capitol	1
caesar	1
caesar	2
caesar	2
did	1
enact	1
hath	1
I	1
I	1
i'	1
it	2
julius	1
killed	1
killed	1
let	2
me	1
noble	2
so	2
the	1
the	2
told	2
you	2
was	1
was	2

term	doc. freq.	→	postings lists
ambitious	1	→	2
be	1	→	2
brutus	2	→	1 → 2
capitol	1	→	1
caesar	2	→	1 → 2
did	1	→	1
enact	1	→	1
hath	1	→	2
i	1	→	1
i'	1	→	1
it	1	→	2
julius	1	→	1
killed	1	→	1
let	1	→	2
me	1	→	1
noble	1	→	2
so	1	→	2
the	2	→	1 → 2
told	1	→	2
you	1	→	2
was	2	→	1 → 2
with	1	→	2

AI Links All

存储开销



稍后的课程会讨论:

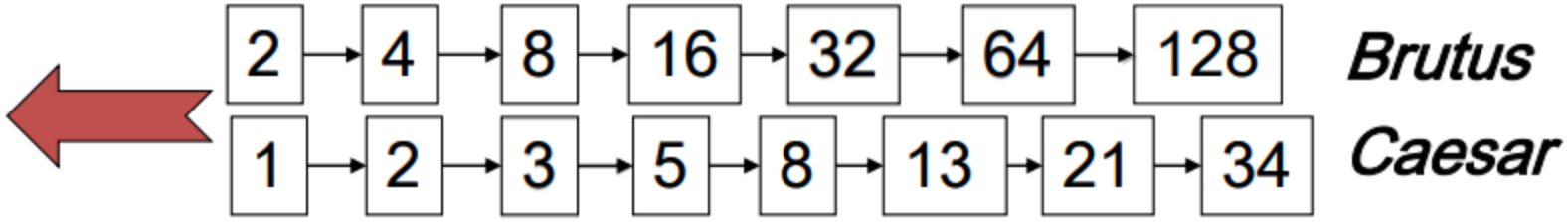
- 如何高效地建立索引?
- 需要多少存储空间？

查询处理：AND

- 考虑处理这样的查询：

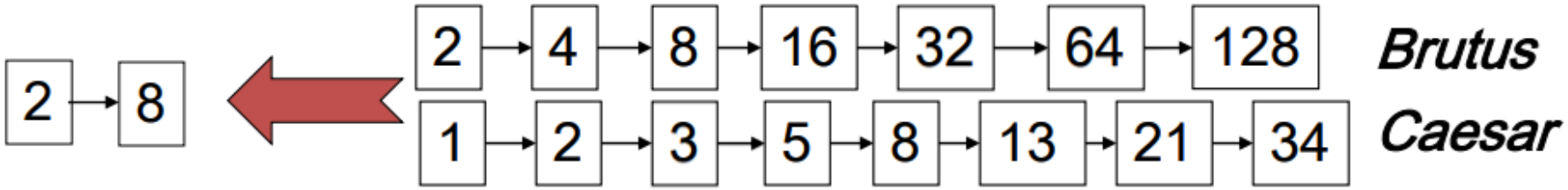
  *Brutus **AND** Caesar*

  - 在字典中找到Brutus，得到它的倒排记录表
  - 在字典中找到Caesar，得到它的倒排记录表
  - 合并两个倒排列表



倒排记录表的合并

- 同时扫描两个倒排记录表，所需时间和倒排记录的数量呈线性关系。



如果倒排记录表的长度分别为x和y，则合并共需 O(*x+y*)次操作
注意：倒排记录已经按照docID排练好了

倒排索引小结

- 倒排索引（Inverted index）
  - 对于每一个词项，存储所有包含这个词项的文档的一个列表
  - 词项 + 倒排记录
- 倒排索引的建立过程
  - 词条化模块 →语言学模块→索引模块
- 查询的处理 ←→ 倒排记录表的“合并”
  - AND

京东智联云 & AI
AI Links All

布尔检索模型

- 文档表示
  - 一个文档被表示为关键词的集合
- 查询表示
  - 查询式(Queries)被表示为关键词的布尔组合，用“与、或、非”连接起来（主析取范式DNF）
- 相关度计算
  - 一个文档当且仅当它能够满足布尔查询式时，才将其检索出来
  - 检索策略是二值匹配

AI Links All

布尔检索模型

布尔检索：精确匹配

- 布尔模型可以用来处理布尔表达式形式的查询
  - 布尔查询使用AND,OR,和NOT来连接查询词汇
    - ◆将文档看作词汇的集合
    - ◆精确：匹配 或 不匹配
  - 布尔模型也许是IR系统中的最简单的模型
  - 是近30年来最主要的商业搜索工具
  - 当前使用的很多系统仍然是使用的布尔模型
    - 电子邮件，图书馆分类系统，mac osx的spotlight

NeuHub AI Links All

布尔检索模型

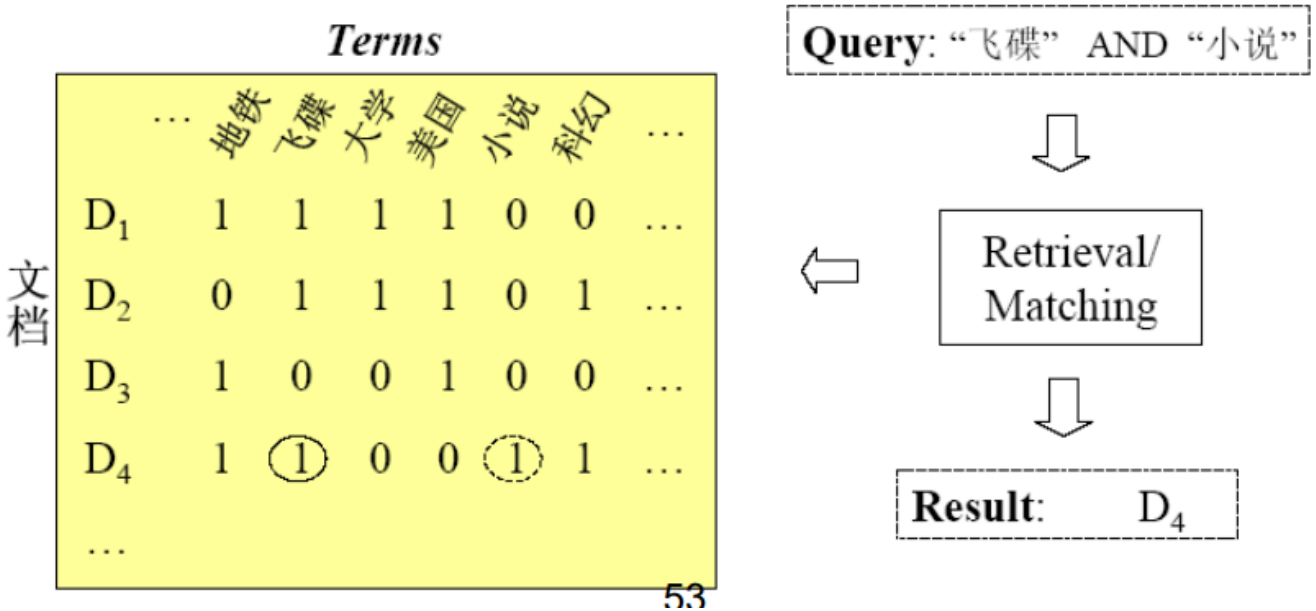
布尔检索应用：关系数据库SQL

- SELECT * FROM Persons WHERE FirstName='Bush'
- SELECT * FROM Persons WHERE Year>1965
- SELECT * FROM Persons WHERE firstname='Thomas ‘
  OR lastname='Carter'
- SELECT * FROM Persons WHERE
  (FirstName='Thomas ‘
  OR FirstName='William') AND LastName='Carter'
- SELECT * FROM Persons
  WHERE LastName IN ('Adams','Carter')
- SELECT * FROM Persons WHERE LastName
  NOT BETWEEN 'Adams' AND 'Carter'
- SELECT * FROM Persons WHERE City LIKE 'N%'

布尔检索模型

示例：精确匹配的欠缺

- “飞碟” AND “小说”：只能检索出D4，无法显现D1,D2,D3的差异
- “飞碟” OR “小说”：可以检出D1,D2,D4，但无法显现它们的差异



贪心科技 | 让每个人享受个性化教育服务

布尔检索模型

示例：WestLaw http://www.westlaw.com/

Westlaw BETTER RESULTS FASTER.

- 最大的收费的法律搜索服务提供商
- 几十T的数据; 700,000多用户
- 大多数用户仍然使用布尔查询
- 查询的例子:
  - What is the statute of limitations in cases involving the federal tort claims act?
  - LIMIT! /3 STATUTE ACTION /S FEDERAL /2 TORT /3 CLAIM
    - /3 = within 3 words, /S = in same sentence
- 许多专业的用户通常喜欢使用布尔查询
  - 因为用户精确的知道自己会得到什么
- 但是这并不意味着能得到更好的结果

京东智联云 & AI
AI Links All

布尔检索模型

- 对于任意的布尔查询组合怎么做？

(Brutus OR Caesar) AND NOT (Antony OR Cleopatra)

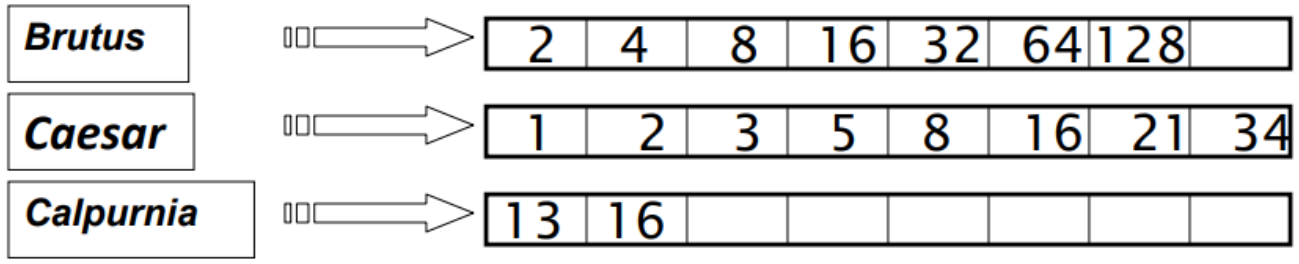
- 依然能够在“线性”时间内完成吗？
  - 和什么呈线性关系？
- 能否做得更好？

NeuHub AI Links All

布尔检索模型

查询的优化

- 处理查询的最佳顺序是什么？
- 考虑一个使用**AND**连接**n**个词汇的查询
- 对于每一个词汇，都得到它的倒排记录表，然后进行“合并”操作



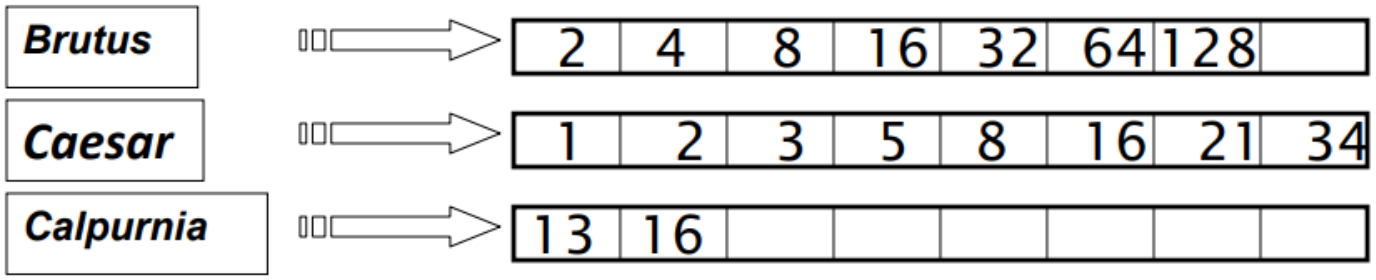
查询: ***Brutus*** *AND* ***Calpurnia*** *AND* ***Caesar***

布尔检索模型

查询优化的例子

- 按照文档频率的顺序进行处理：
  - 先处理文档频率小的，再处理大的

这就是为什么我们前面提到要存储词条的文档频率



按照(*Calpurnia AND Brutus*) *AND Caesar*的顺利处理查询

布尔检索模型

更一般的优化

- e.g., *(madding OR crowd) AND (ignoble OR strife) AND (killed OR slain)*
- 获得所有词项的文档频率
- 保守地估计出每个OR操作后的结果大小
- 按照结果从小到大的顺序执行AND

布尔检索模型

- 请推荐如下查询的处理次序

(tangerine OR trees) AND
(marmalade OR skies) AND
(kaleidoscope OR eyes)

Term	Freq
eyes	213312
kaleidoscope	87009
marmalade	107913
skies	271658
tangerine	46653
trees	316812

Links All

布尔检索模型

除了词查询外，信息检索还有什么问题？

- 词组？
  - 中国科学技术大学
- 词项邻近: 在“合肥”附近的“老母鸡”.
  - 对文档建索引的时候需要考虑词出现的位置
- 文档中的区域:
  - 查找*作者*是“D.Manning”，正文含有“Information”的文档

布尔检索及倒排索引小结

- 一个简单的例子
  - 词项文档矩阵
- 倒排索引
  - 索引建立、查询处理
- 布尔检索模型
  - 应用、缺点、优化

NeuHub
AI Links All

提纲

- 信息检索概述
- 布尔检索及倒排索引
- **信息检索模型综述**

- 索引压缩与VB编码
- 查询优化与WAND算法

- 信息检索系统评估
- 实战练习

京东智联云 & AI
贪心科技 GREEDY TECHNOLOGY
NeuHub AI Links All

相关性度量

- 确定文档和查询之间的相关度是IR的核心问题

Query:

Bill Clinton

Document:

The White House
PRESIDENT GEORGE W. BUSH
William J. Clinton
During the administration of William Jefferson Clinton, the U.S. enjoyed more peace and economic well being than at any time in its history. He was the first Democratic president since Franklin D. Roosevelt to win a second term. He could point to the lowest unemployment rate in modern times, the lowest inflation in 30 years, the highest home ownership in the country's history, dropping crime rates in many places, and reduced welfare rolls. He proposed the first balanced budget in decades and achieved a budget surplus. As part of a plan to celebrate the millennium in 2000, Clinton called for a great national initiative to end racial discrimination.
After the failure in his second year of a huge program of health care reform, Clinton shifted emphasis, declaring "the era of big government is over." He sought legislation to upgrade education, to protect jobs of parents who must care for sick children, to restrict handgun sales, and to strengthen environmental rules.

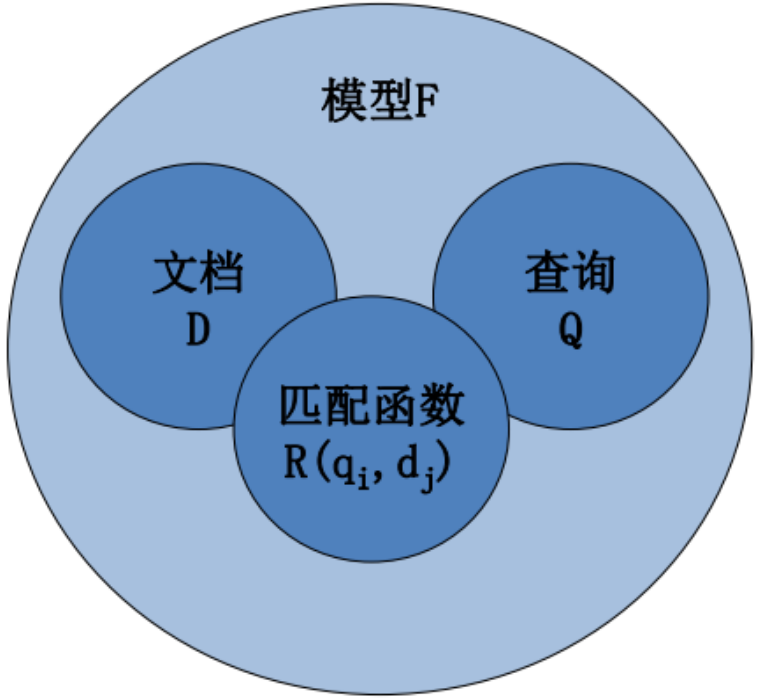
Relevant?

How relevant?

NeuHub
AI Links All

检索模型的定义

- 信息检索模型是描述信息检索中的文档、查询和它们之间的关系(匹配函数)的数学模型。



NeuHub
AI Links All

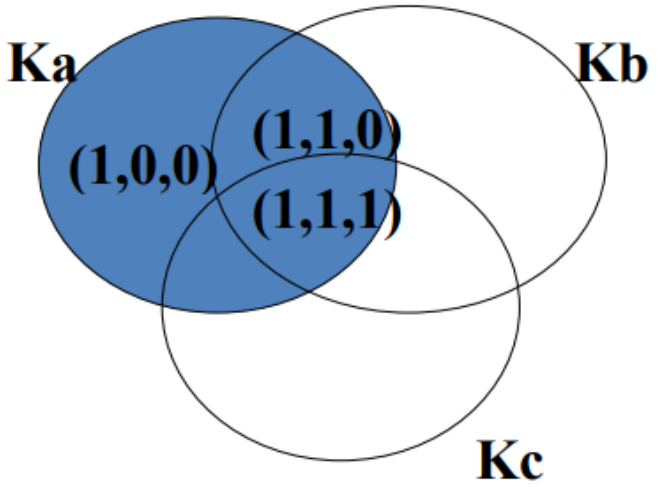
信息检索模型

- 基于文本内容的检索模型
  - 布尔模型
  - 向量空间模型
  - 概率模型
  - 统计语言模型
- 与内容无关的其他检索模型
  - 基于链接分析的模型
  - 基于关联的模型

NeuHub AI Links All

基于内容：布尔模型

- 文档表示
  - 一个文档被表示为关键词的集合
- 查询表示
  - 查询式(Queries)被表示为关键词的布尔组合，用“与、或、非”连接起来（主析取范式DNF）
- 相关度计算
  - 一个文档当且仅当它能够满足布尔查询式时，才将其检索出来
  - 检索策略是二值匹配



$$q = k_a \wedge (k_b \vee \neg k_c)$$

$$q_{dnf} = (1,1,1) \vee (1,1,0) \vee (1,0,0)$$

AI Links All

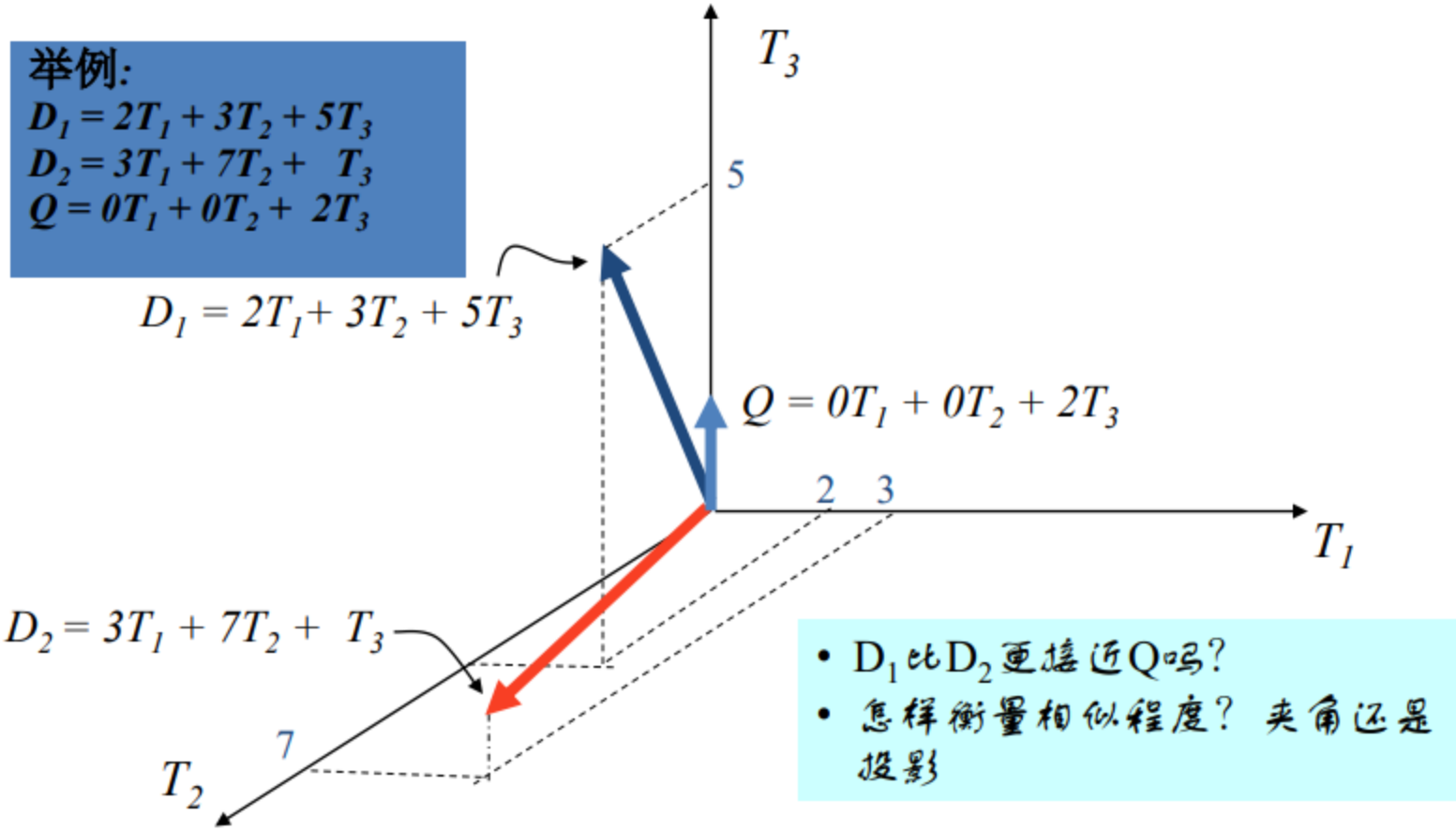
基于内容：布尔模型

- 优点
  - 由于查询简单，因此容易理解
  - 通过使用复杂的布尔表达式，可方便地控制查询结果
  - 相当有效的实现方法
  - 经过某种训练的用户可以容易地写出布尔查询式
  - 布尔模型可以通过扩展来包含排序的功能
- 缺点
  - 弱。不支持部分匹配，完全匹配会导致结果太多或太少
  - 非常刚性：“与”意味着全部；“或”意味着任何一个
  - 原则上讲，所有被匹配的文档都将被返回
  - 不考虑索引词的权重，所有文档都以相同的方式和查询相匹配
  - 很难进行自动的相关反馈
  - 如果一篇文档被用户确认为相关或者不相关，怎样相应地修改查询式呢？

基于内容：向量空间模型

- 文档表示
  - 一个文档被表示为关键词构成的向量。
  - 一个文档D就可以表示为$D(t_1,t_2,\ldots,t_n)$，其中n就代表了关键字的数量。
  - 特征项权重Wk（Term Weight）：指特征项t_n能够代表文档D能力的大小，体现了特征项在文档中的重要程度。
- 查询表示
  - 一个文档被表示为关键词构成的向量$Q(t_1,t_2,\ldots,t_n)$。
- 相关度计算
  - 向量间的距离：$D(t_1,t_2,\ldots,t_n)$，$Q(t_1,t_2,\ldots,t_n)$

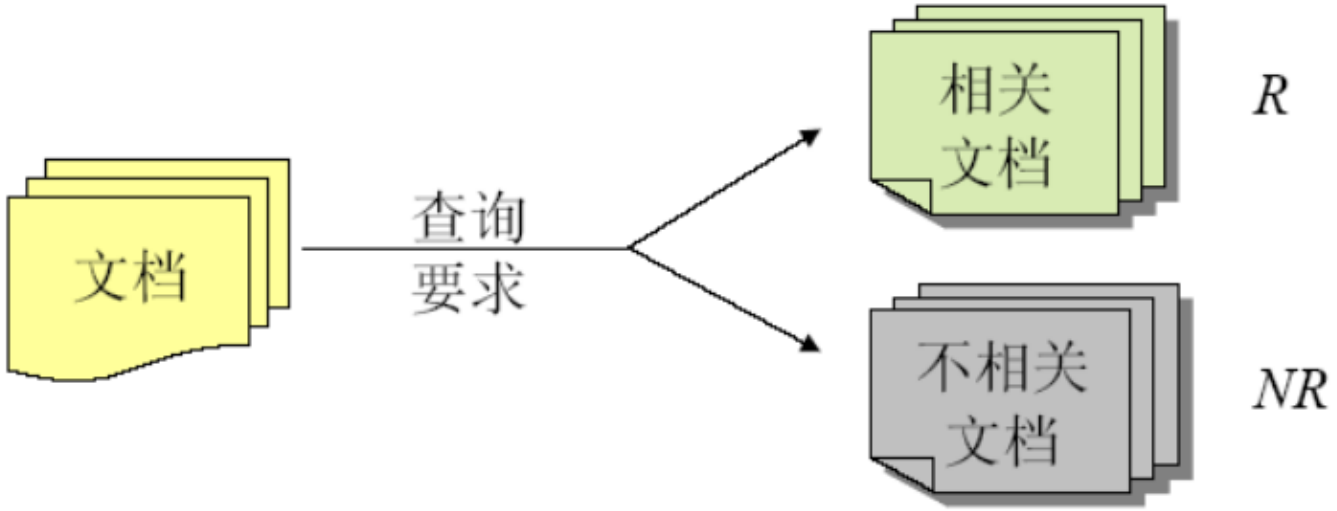
基于内容：向量空间模型


举例:
$D_1 = 2T_1 + 3T_2 + 5T_3$
$D_2 = 3T_1 + 7T_2 + T_3$
$Q = 0T_1 + 0T_2 + 2T_3$
T_3
5
$D_1 = 2T_1 + 3T_2 + 5T_3$
$Q = 0T_1 + 0T_2 + 2T_3$
2
3
T_1
$D_2 = 3T_1 + 7T_2 + T_3$
7
T_2
• D_1比D_2更接近Q吗?
• 怎样衡量相似程度? 夹角还是投影

基于内容：向量空间模型

- 特点
  - 基于关键词(一个文本由一个关键词列表组成)
  - 根据关键词的出现频率计算相似度
    - 例如：文档的统计特性
  - 用户规定一个词项(term)集合，可以给每个词项附加权重
    - 未加权的词项: Q = < database; text; information >
    - 加权的词项: Q = <database 0.5; text 0.8; information 0.2 >
    - 查询式中没有布尔条件
  - 根据相似度对输出结果进行排序
  - 支持自动的相关反馈
    - 有用的词项被添加到原始的查询式中
    - 例如：Q = <database; text; information; document >
- 三个关键问题
  - 怎样确定文档中哪些词是重要的词？（索引项）
  - 怎样确定一个词在某个文档中或在整个文档集中的重要程度？（权重）
  - 怎样确定一个文档和一个查询式之间的相似度？（相关度）

基于内容：概率模型



检索问题即求条件概率问题

If Prob($R|d_i, q$) > Prob($NR|d_i, q$) then d_i是检索结果，
否则不是检索结果

基于内容：统计语言模型

【基于内容】统计语言模型（Probabilistic Language Models）

- “语言”就是其字母表上的某种概率分布,该分布反映了任何一个字母序列成为该语言的一个句子(或其他任何的语言单元) 的可能性，称这个概率分布为语言模型。
- 给定的一个语言，对于一个语言“句子”（符号串），可以估计其出现的概率。

- **P(南京市长)P(江大桥|南京市长) <<**
  P(南京市)P(长江大桥|南京市)

- **P(中国科技大学) >> P(中国大学科技)**

AI Links All

内容无关：链接分析模型

- 对于超文本（例如WWW上的网页），超链结构是个非常丰富和重要的资源，如果能够充分利用的话，可以极大地提高检索结果的质量。

- Sergey Brin 和Larry Page 在1998 年提出了PageRank 算法
- J.Kleinberg 于1998年提出了HITS 算法
- 其它一些学者也相继提出了另外的链接分析算法，如SALSA，PHITS，Bayesian等算法。

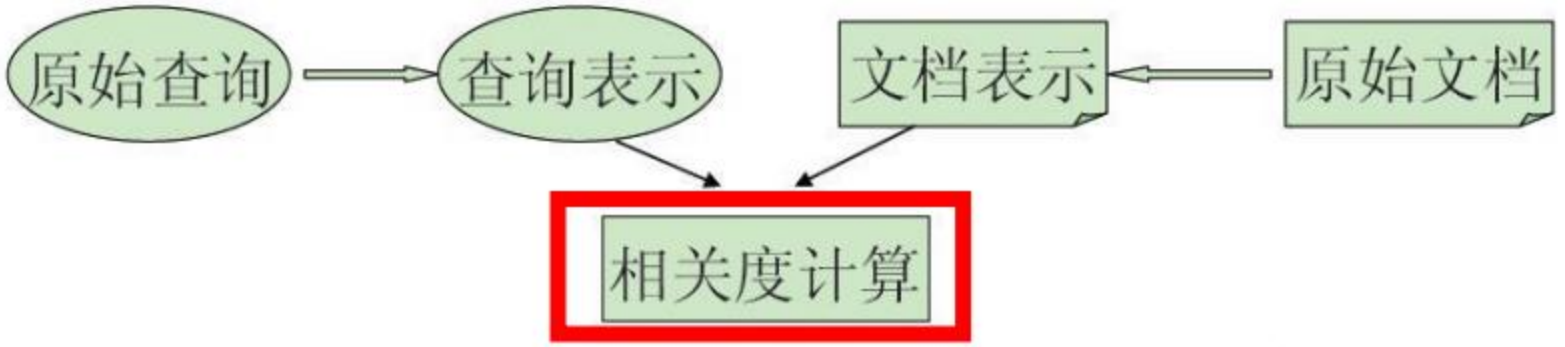
内容无关：基于关联的模型

- 基本思想：同一个类型下的两个对象,如果经常连接到相同的其他对象，那么这两个对象的相似性应该很高。

- 示例（ Simrank算法）
  - 1.利用文章的相互之间的引用关系计算文本的相似度。
    - ----两个文档的引文相同，那么这两个文档的相似性很高。
  - 2.利用文章的一些外部信息(关联)计算文本的相似度。
    - ---文档外部信息（作者，发表会议）
    - ---两个文档有共同的作者，发表到共同的会议上，那么这两个文档的相似度很高。

信息检索小结

- 信息检索模型是指如何对查询和文档进行表示，然后对它们进行相似度计算的框架和方法
  - 本质上是对相关度建模
- 信息检索模型是IR中的核心内容之一



AI Links All

信息检索小结

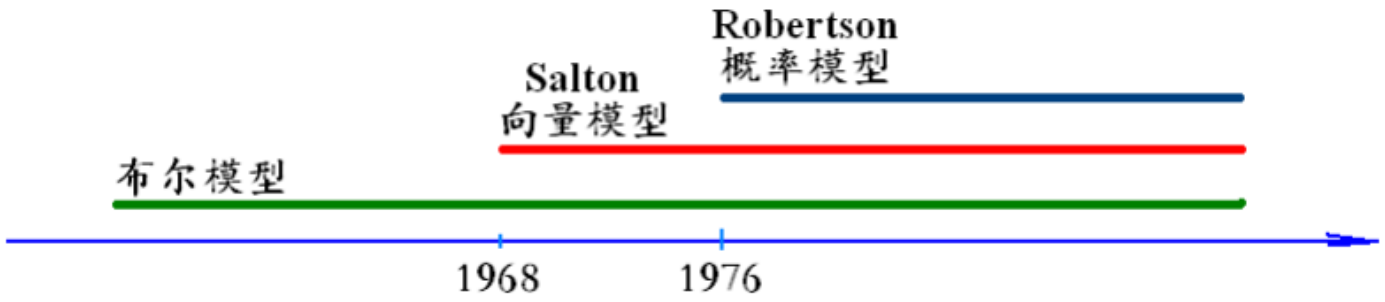
- 核心问题：相关度的计算
- 查询的表示/文档的表示
  - 关键字的集合？
  - 关键字构成的向量？
  - 关键字的重要程度？
  - 文档可视为关键字的集合？
  - 文档中单词出现的概率？

- 术语集合（representative terms）
- 标引词（index term）
- 标引项的权重（Weight）

NeuHub
AI Links All

信息检索小结

- 集合论模型 (Set Theoretic models)
  - 布尔模型（ Boolean Model, BM）、模糊集合模型、扩展布尔模型
- 代数模型 (Algebraic models)
  - 向量空间模型（Vector Space Model, VSM）、广义向量空间模型、潜在语义标引模型、神经网络模型
- 概率模型 (Probabilistic models)
  - 经典概率论模型（PM）、推理网络模型、置信网络模型



小结

- 信息检索概述
  - 定义、发展历史
- 布尔检索及倒排索引
  - 信息检索例子、倒排索引、布尔查询模型
- 信息检索模型综述
  - 基于内容、内容无关

贪心科技 GREEDY TECHNOLOGY
NeuHub AI Links All